

Resolución Práctica 2.2: Aplicación *Echo* usando TCP

Fichero “*echo_cli_tcp.py*”:

```
#!/usr/bin/env python3

import socket, sys

PORT = 50002

if len( sys.argv ) != 2:
    print( "Uso: {} <servidor>".format( sys.argv[0] ) )
    exit( 1 )

dir_serv = (sys.argv[1], PORT)

s = socket.socket( socket.AF_INET, socket.SOCK_STREAM )
s.connect( dir_serv )

print( "Introduce el mensaje que quieres enviar (mensaje vacío para
terminar):" )
while True:
    mensaje = input()
    if not mensaje:
        break
    s.sendall( mensaje.encode() )
    # Es necesario un bucle porque no hay garantías de que la respuesta
    # completa se reciba en una única lectura.
    bytes_por_leer = len( mensaje.encode() )
    mensaje = b""
    while bytes_por_leer:
        buf = s.recv( bytes_por_leer )
        mensaje += buf
        bytes_por_leer -= len( buf )
    print( mensaje.decode() )
s.close()
```

Fichero “*echo_ser_tcp.py*”:

```
#!/usr/bin/env python3

import socket

PORT = 50002

s = socket.socket( socket.AF_INET, socket.SOCK_STREAM )

s.bind( ('', PORT) )
s.listen( 5 )

while True:
    dialogo, dir_cli = s.accept()
    print( "Cliente conectado desde {}:{}".format( dir_cli[0], dir_cli[1] ) )
    while True:
        buf = dialogo.recv( 1024 )
        if not buf:
            break
        dialogo.sendall( buf )
        print( "Solicitud de cierre de conexión recibida." )
        dialogo.close()
s.close()
```